☐ **Generate Collection** **Print**

L4: Entry 1 of 1   File: PGPB   Mar 27, 2003

DOCUMENT-IDENTIFIER: US 20030058277 A1
TITLE: A VIEW CONFIGURER IN A PRESENTATION SERVICES PATTERNS ENVIROMENT

Pre-Grant Publication (PGPub) Document Number:
20030058277

Detail Description Paragraph:
[3584] When the Server has retrieved all of the data meeting the search criteria,
the Server builds the last "page." When the last page is returned to the client,
the "last found key" is left blank. This notifies the client the search is complete
and no more data matching the search exists on the Server. Note that the last page
is usually smaller than the other pages.

Detail Description Paragraph:
[4097] The examples have shown a single exception handler being used. In practice
it is more likely that multiple will be used. For example, the exception handler on
a server may have different requirements or constraints than a client, or one
client may be GUI based and display pop-up error messages, where another client is
a batch program that needs to send notification messages to Operations. This can be
handled by creating multiple handlers or using the Strategy pattern to customize
the behavior.

☐ Generate Collection   Print

L34: Entry 1 of 1   File: USPT   Jul 29, 2003

US-PAT-NO: 6601234
DOCUMENT-IDENTIFIER: US 6601234 B1
** See image for **Certificate of Correction** **

TITLE: Attribute dictionary in a business logic services environment

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Accenture LLP	Palo Alto	CA			02

APPL-NO: 09/ 388022   [PALM]
DATE FILED: August 31, 1999

INT-CL: [07] G06 F 9/44

US-CL-ISSUED: 717/108; 717/107, 717/116, 705/7
US-CL-CURRENT: 717/108; 705/7, 717/107, 717/116

FIELD-OF-SEARCH: 705/7-11, 717/107, 717/109

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected   Search ALL   Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5047918	September 1991	Schwartz et al.	707/203
☐	5133075	July 1992	Risch	707/201
☐	5187787	February 1993	Skeen et al.	709/314
☐	5241580	August 1993	Babson, III	379/15
☐	5291593	March 1994	Abraham et al.	707/103
☐	5301270	April 1994	Steinberg et al.	345/326
☐	5301320	April 1994	McAttee et al.	395/650

☐	5313636	May 1994	Noble et al.	707/1
☐	5414812	May 1995	Filip et al.	707/103
☐	5434978	July 1995	Dockter et al.	709/230
☐	5437038	July 1995	Silberbauer et al.	395/700
☐	5457797	October 1995	Butterworth et al.	709/302
☐	5463686	October 1995	Lebourges	379/220
☐	5471629	November 1995	Risch	707/201
☐	5475844	December 1995	Shiramizu et al.	709/104
☐	5499371	March 1996	Henninger et al.	717/2
☐	5560005	September 1996	Hoover et al.	707/10
☐	5568644	October 1996	Nelson et al.	395/741
☐	5581758	December 1996	Burnett et al.	707/103
☐	5606664	February 1997	Brown et al.	709/224
☐	5623418	April 1997	Rostoker et al.	716/1
☐	5642511	June 1997	Chow et al.	395/701
☐	5649139	July 1997	Weinreb et al.	707/202
☐	5671386	September 1997	Blair et al.	395/405
☐	5675748	October 1997	Ross	395/284
☐	5677997	October 1997	Talatik	706/45
☐	5680602	October 1997	Bloem et al.	707/1
☐	5692107	November 1997	Simoudis et al.	706/12
☐	5706506	January 1998	Jensen et al.	707/103
☐	5708828	January 1998	Coleman	395/785
☐	5710901	January 1998	Srodghill et al.	345/339
☐	5715397	February 1998	Ogawa et al.	395/200.18
☐	5721908	February 1998	Lagarde et al.	
☐	5724575	March 1998	Hoover et al.	707/10
☐	5732263	March 1998	Havens et al.	707/103
☐	5732270	March 1998	Foody et al.	709/303
☐	5737607	April 1998	Hamilton et al.	395/701
☐	5751965	May 1998	Mayo et al.	709/224
☐	5758351	May 1998	Gibson	707/104
☐	5761513	June 1998	Yellin et al.	395/705
☐	5764235	June 1998	Hunt et al.	345/428
☐	5764955	June 1998	Doolan	709/223
☐	5774660	June 1998	Brendel et al.	709/201
☐	5778368	July 1998	Hogan et al.	707/10
	5787413	July 1998	Kauffman et al.	707/2

☐				
☐	5799310	August 1998	Anderson et al.	707/102
☐	5867153	February 1999	Grandcolas et al.	345/326
☐	5870742	February 1999	Chang et al.	707/8
☐	5870746	February 1999	Knutson et al.	707/101
☐	5872973	February 1999	Mitchell et al.	709/332
☐	5873086	February 1999	Fujii et al.	707/10
☐	5878408	March 1999	Van Huben et al.	707/1
☐	5890133	March 1999	Ernst	705/7
☐	5892909	April 1999	Grasso et al.	709/201
☐	5896383	April 1999	Wakeland	370/400
☐	5898870	April 1999	Okuda et al.	709/104
☐	5905873	May 1999	Hartmann et al.	395/200.79
☐	5905897	May 1999	Chan et al.	395/733
☐	5907704	May 1999	Gudmundson et al.	395/701
☐	5909540	June 1999	Carter et al.	714/4
☐	5915115	June 1999	Talati	717/5
☐	5920703	July 1999	Campbell et al.	395/200.66
☐	5933816	August 1999	Zeannah et al.	705/35
☐	5940075	August 1999	Mutschler, III et al.	345/335
☐	5940594	August 1999	Ali et al.	709/203
☐	5946694	August 1999	Copeland et al.	707/103
☐	5946697	August 1999	Shen	707/104
☐	5953707	September 1999	Huang et al.	705/10
☐	5958012	September 1999	Battat et al.	709/224
☐	5960200	September 1999	Eager et al.	717/5
☐	5966451	October 1999	Utsumi	380/49
☐	5987247	November 1999	Lau	717/2
☐	5987501	November 1999	Hamilton et al.	709/203
☐	5987514	November 1999	Rangarajan	709/224
☐	5987633	November 1999	Newman et al.	714/712
☐	5995753	November 1999	Walker	717/2
☐	5995945	November 1999	Notani et al.	705/28
☐	5995975	November 1999	Malcolm	707/103
☐	5999948	December 1999	Nelson	
☐	6006230	December 1999	Ludwig et al.	707/10
☐	6016394	January 2000	Walker	717/1
	6018743	January 2000	Xu	707/103R

☐				
☐	6023722	February 2000	Colyer	709/201
☐	6029174	February 2000	Sprenger et al.	707/103
☐	6029177	February 2000	Sadiq et al.	707/201
☐	6032153	February 2000	Sadiq et al.	707/103
☐	6035303	March 2000	Baer et al.	707/103
☐	6038598	March 2000	Danneels	709/219
☐	6041365	March 2000	Kleinerman	709/302
☐	6052739	April 2000	Bopardikar et al.	709/301
☐	6057856	May 2000	Miyashita et al.	345/435
☐	6070191	May 2000	Narendran et al.	709/226
☐	6078960	June 2000	Ballard	709/229
☐	6081837	June 2000	Stedman et al.	709/219
☐	6083276	July 2000	Davidson et al.	717/1
☐	6085198	July 2000	Skinner et al.	707/103
☐	6092118	July 2000	Tsang	709/246
☐	6108703	August 2000	Leighton et al.	709/226
☐	6115752	September 2000	Chauhan	709/241
☐	6125359	September 2000	Lautzenheiser et al.	706/60
☐	6128279	October 2000	O'Neil et al.	370/229
☐	6141660	October 2000	Bach et al.	345/352
☐	6141759	October 2000	Braddy	713/201
☐	6144991	November 2000	England	709/205
☐	6148335	November 2000	Haggard et al.	709/224
☐	6148361	November 2000	Carpenter et al.	710/260
☐	6154212	November 2000	Eick et al.	345/356
☐	6157940	December 2000	Marullo et al.	709/22
☐	6182182	January 2001	Bradley et al.	710/129
☐	6202099	March 2001	Gillies et al.	709/319
☐	6223209	April 2001	Watson	709/201
☐	6243761	June 2001	Mogul et al.	709/246

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0123456	January 2000	EP	100/100
WO92/01251	January 1992	WO	
WO 99/08208	February 1999	WO	
WO 99/44155	September 1999	WO	

PCT/US00/23885	August 2000	WO
PCT/US00/23999	August 2000	WO
PCT/US00/24082	August 2000	WO
PCT/US00/24083	August 2000	WO
PCT/US00/24084	August 2000	WO
PCT/US00/24085	August 2000	WO
PCT/US00/24086	August 2000	WO
PCT/US00/24125	August 2000	WO
PCT/US/00/24188	August 2000	WO
PCT/US00/24189	August 2000	WO
PCT/US00/24236	August 2000	WO

OTHER PUBLICATIONS

Stefik, M. Bobrow, D.G., and Kahn, K. Access-oriented programming for a
multiparadigm environment. Proceedings of the Hawaii International Conference on
System Sciences, Jan. 1986.*
Lutz, Mark. Programming Python. O'Reilly & Associates, 1996, pp. 415-416,
downloaded Jul. 15, 2002 from NetLibrary.com.*
Gamma, Erich et al. Design Patterns: Elements of Resuable Object-oriented Software.
Addison-Wesley, 1994, pp. 185-193.*
Rosenthala, James W. et al. A Fine-grained Access Control Model for Object-oriented
DBMSs. First published in IFIP Transactions: Database Security VII, Elsevier,
1995.*
Wang, Haojin. "Managed Object Class Dictionary" in Telecommunications Network
Management, McGraw-Hill, 1999, Appendix D. Retrieved from Books24x7.com on Jul. 15,
2002.*
Kinexis. Object-orientation and Transaction Processing: Where Do They Meet. OOPSLA
Keynote, Oct. 6-11, 1991.*
Lee et al. Path Dictionary: A New Access Method for Query Processing in Object-
oriented Databases. IEEE Transactions on Knowledge and Data Engineering, v10, n3,
May/Jun. 1998.*
Buddrus et al. Enacting Authorization Models for Object-oriented Databases.
Database and Expert Systems Applications, Proceedings., Seventh International
Workshop, Sep. 9-10, 1996, pp. 116-121.*
Bertino et al. Trigger Inheritance and Overriding in an Active Object Database
System. IEEE Transactions on Knowledge and Data Engineering, v12, n4, Jul./Aug.,
2000.*
Kovalerchuck et al., comparison of relational methods and attribute-based methods
for data mining in intelligent systems, proceedings of the 1999 IEEE, International
Symposium on Intelligent Systems and Semiotics, Cambridge, MA, PP 162-166. Date
Sep. 1999.
ANSII Standard for the Programming Language C++, First Edition ISO/IEC 14882: 1998.
Date Sep. 1998.
The Annotated C++ Reference Manual ANSII Base Document, M.A. Ellis and B.
Stroustrup. Date Jul. 1990.
IBM Dictionary of Computing, pp. 140, 241, 299, 728.
Microsoft Corporation, Microsoft Solutions Framework Overview A Quick Tour of the
MSF Models, URL: http://channels.microsoft.com/enterprise/support/support/consult,
Viewed Oct. 9, 1999.
Microsoft Corporation, Microsoft Solutions Framework Overview A Quick Tour of the
MSF Models, URL: http://channels.microsoft.com/enterprise/support/support/consult,
Viewed Oct. 9, 1999.

ART-UNIT: 3623

PRIMARY-EXAMINER: Hafiz; Tariq R.

ASSISTANT-EXAMINER: Jeanty; Romain

ATTY-AGENT-FIRM: Oppenheimer Wolff & Donnelly LLP

ABSTRACT:

A system and method are provided for controlling access to data of a business
object via an attribute dictionary. The attribute dictionary, which stores
attribute names and values, is dispatched over a network. A helper facade is
provided for interfacing a business object and the attribute dictionary. Next, it
is verified that a current user is authorized to either set or get one of the
attribute values upon a request which includes the attribute name that corresponds
to the attribute value. The helper facade is called to set, get, or update one of
the attribute values based on the corresponding attribute name, wherein the helper
facade shields the attribute dictionary from the application code of the business
object. The attribute value in the attribute dictionary is obtained or updated if
the verification is successful, and a dirty flag is set in the attribute dictionary
and an indicator is broadcast upon the attribute value being updated.

15 Claims, 195 Drawing figures

☐   ⠆⠆⠆Generate Collection ⠄⠆⠄ Print

L39: Entry 1 of 1   File: USPT   Jul 29, 2003

DOCUMENT-IDENTIFIER: US 6601234 B1
** See image for Certificate of Correction **
TITLE: Attribute dictionary in a business logic services environment

Brief Summary Text (10):
The architecture of the Web follows a conventional client-server model. The terms
"client" and "server" are used to refer to a computer's general role as a requester
of data (the client) or provider of data (the server). Under the Web environment,
Web browsers reside in clients and Web documents reside in servers. Web clients and
Web servers communicate using a protocol called "HyperText Transfer
Protocol" (HTTP). A browser opens a connection to a server and initiates a request
for a document. The server delivers the requested document, typically in the form
of a text document coded in a standard Hypertext Markup Language (HTML) format, and
when the connection is closed in the above interaction, the server serves a passive
role, i.e., it accepts commands from the client and cannot request the client to
perform any action.

Detailed Description Text (300):
Web Browsers employ standard protocols such as Hypertext Transfer Protocol (HTTP)
and File Transfer Protocol (FTP) to provide seamless access to documents across
machine and network boundaries.

Detailed Description Text (501):
TelAlert; e-mail systems e-mail systems--some e-mail systems and fax servers can be
configured to generate pages to notify users when a defined event occurs such as e-
mail/fax arriving. Telamon's TetAlert--TelAlert provides notification capabilities
for UNIX systems. For example, it can page support personnel in the event of system
problems.

Detailed Description Text (504):
Phone services enable clients, servers, and specialized telephony nodes (PBXs,
ACDs, etc.) to control the telephony environment through the following telephony
controls: Call control Controls telephone features Controls recorded messages
Manipulates real time call activities (e.g., make call, answer, transfer, hold,
conference, mute transfer, release, route call, call treatments and digits
collected) Telephone status control Controls telephone status functions Logs users
in and out of the system Sets ready, not ready, and make busy statuses for users

Detailed Description Text (600):
File Transfer services enable the sending and receiving of files or other large
blocks of data between two resources. In addition to basic file transport, features
for security, guaranteed delivery, sending and tracking sets of files, and error
logging may be needed if a more robust file transfer architecture is required. The
following are examples of File Transfer standards: File Transfer Protocol (FTP)
allows users to upload and download files across the network. FTP also provides a
mechanism to obtain filename, directory name, attributes and file size information.
Remote file access protocols, such as Network File System (NFS) also use a block
transfer method, but are optimized for online read/write paging of a file.

HyperText Transfer Protocol (HTTP)--Within a Web-based environment, Web servers transfer HTML pages to clients using HTTP. HTTP can be thought of as a lightweight file transfer protocol optimized for transferring small files. HTTP reduces the inefficiencies of the FTP protocol. HTTP runs on top of TCP/IP and was developed specifically for the transmission of hypertext between client and server. The HTTP standard is changing rapidly. Secure Hypertext Transfer Protocol (S-HTTP)--a secure form of HTTP, mostly for financial transactions on the Web. S-HTTP has gained a small level of acceptance among merchants selling products on the Internet as a way to conduct financial transactions (using credit card numbers, passing sensitive information) without the risk of unauthorized people intercepting this information. S-HTTP incorporates various cryptographic message formats such as DSA and RSA standards into both the Web client and the Web server. File Transfer and Access Management (FTAM)--The OSI (Open Systems Interconnection) standard for file transfer, file access, and file management across platforms.

Detailed Description Text (607):
The following product provides File Transfer translation: IBM's Files On-Demand gateway--acts as a gateway between Web-based and mainframe-based FTP services to allow users to download mainframe-based files from a World Wide Web browser.

Detailed Description Text (654):
X.400--The X.400 message handling system standard defines a platform independent standard for store-and-forward message transfers among mail servers. X.400 is often used as a backbone e-mail service, with gateways providing interconnection with end-user systems.

Detailed Description Text (655):
SMTP--Simple Mail Transfer Protocol (SMTP) is a UNIX/Internet standard for transferring e-mail among servers.

Detailed Description Text (758):
Encryption that is architected into Web-based solutions Netscape's Secure Sockets Layer (SSL)--provides encryption for World Wide Web browsers. S-HTTP--a secure version of the HTTP data transfer standard; used in conjunction with the World Wide Web.

Detailed Description Text (2260):
When the Server has retrieved all of the data meeting the search criteria, the Server builds the last "page." When the last page is returned to the client, the "last found key" is left blank. This notifies the client the search is complete and no more data matching the search exists on the Server. Note that the last page is usually smaller than the other pages.

Detailed Description Text (2642):
The examples have shown a single exception handler being used. In practice it is more likely that multiple will be used. For example, the exception handler on a server may have different requirements or constraints than a client, or one client may be GUI based and display pop-up error messages, where another client is a batch program that needs to send notification messages to Operations. This can be handled by creating multiple handlers or using the Strategy pattern to customize the behavior.

☐ ▐ Generate Collection ▌ ▐ Print ▌

L27: Entry 3 of 3   File: USPT   Jul 1, 2003

DOCUMENT-IDENTIFIER: US 6587127 B1
TITLE: Content player method and server with user profile

Detailed Description Text (103):
It is noted that a personal notification message can be sent to an end user without
use of the aforementioned notification prompt. In this case, a notification message
is sent to the end user for an event or a program which may be of interest based
upon listening behavior or other activity of the end user. Hence, in general, the
server 102 can provide personal notification for content that is broadcast at a
scheduled time based upon either an expressed interest of the end user (e.g.
through the aforementioned active marking for content using the notification
prompt) or a passive interest of the end user (e.g. through monitoring of user
activity).

L33: Entry 2 of 7   File: USPT   Oct 5, 2004

DOCUMENT-IDENTIFIER: US 6801225 B1
TITLE: Data storage in ole systems

Application Filing Date (1):
19990729

Detailed Description Text (36):
As a further level of detail, before the render manager issues a re-render command
to the effects server, the render manager issues a "prepare to render" message
specifying which image in the sequence is to be rendered. The effects server
responds with a notification of its "dependencies", i.e. those rendered images
which are essential before the request by the render manager can be executed. These
might be images rendered by another effect (e.g. the or an immediately preceding
effect in the directed acyclic graph) or images rendered by that effect itself.
This latter case can occur in the example of a motion tracker, where in order to
render, say, image 5, the motion tracker needs its own rendered output for image 4.

Detailed Description Text (43):
As a further extension, the same protocol can be used so that each effects server
can notify the render manager if its output is the same between adjacent images. A
simple example of this is a (fixed) parameter plug-in, where the output is
invariant. A further example is any other effect where the outputs have already
been prepared and cached, so that a straightforward detection can be made as to
whether successive outputs are identical. In response to such a notification, the
render manager pass the information on to an effects server which is later in the
directed acyclic graph. That effects server can then (if appropriate) render only
one of a range of images and repeat that output for other images where its input
remains the same.

Current US Cross Reference Classification (1):
715/765

      L33: Entry 4 of 7   File: USPT   Dec 14, 1999

US-PAT-NO: 6003067
DOCUMENT-IDENTIFIER: US 6003067 A
** See image for Certificate of Correction **

TITLE: Data transmission controlling method and data transmission controlling
system, and computer memory product

DATE-ISSUED: December 14, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Suzuki; Toshimitsu	Kawasaki			JP
Saito; Kazumi	Kawasaki			JP
Yashiro; Sadao	Kawasaki			JP
Muramoto; Takahide	Kawasaki			JP

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Fujitsu Limited	Kawasaki			JP	03

APPL-NO: 08/ 880539   [PALM]
DATE FILED: June 23, 1997

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	9-019477	January 31, 1997

INT-CL: [06] G06 F 15/16

US-CL-ISSUED: 709/204; 709/205, 345/330, 345/340, 345/344, 345/345
US-CL-CURRENT: 709/204; 709/205, 715/753, 715/790, 715/803

FIELD-OF-SEARCH: 395/200.33, 395/200.34, 395/200.54, 345/345, 345/419, 345/473,
345/329, 345/330, 345/344, 709/204, 709/205

PRIOR-ART-DISCLOSED:

                         U.S. PATENT DOCUMENTS

          [Search Selected] [Search ALL] [Clear]

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4833535	May 1989	Ozeki et al.	

☐	5481297	January 1996	Cash et al.	
☐	5583980	December 1996	Anderson	345/473
☐	5682166	October 1997	Takahashi	345/419
☐	5812132	September 1998	Goldstein	345/345

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
3-16390	January 1991	JP	

OTHER PUBLICATIONS

Ambras et al., "MicroScope: an integrated program analysis toolset.", Hewlett-Packard Journal, v39 No. 4, p. 71, Aug. 1988.
Anonymous, "Data Based Advisor COMDEX sneak preview", Data Based Advisor, v7 No. 11, p. 71, Nov. 1989.
Maestri, George, "Aldus AfterImage", Digital Video Magazine, v2 No. 11, p. 85, Dec. 1994.
Richter, Jeffrey, "Extend the Windows 95 shell with application desktop toolbars", Microsoft Systems Journal, v11 No. 3, p. 35, Mar. 1996.

ART-UNIT: 276

PRIMARY-EXAMINER: Asta; Frank J.

ASSISTANT-EXAMINER: Romero; Almari

ATTY-AGENT-FIRM: Staas & Halsey LLP

ABSTRACT:

A data transmission processing method and system and a computer memory product capable of quickly performing the displaying operation of the screen data by transmitting of the screen data except the units which do not display on the screen due to hiding, to the first information processing apparatus from the second information processing apparatus.

23 Claims, 18 Drawing figures

☐ | Generate Collection | | Print |

L33: Entry 4 of 7   File: USPT   Dec 14, 1999

DOCUMENT-IDENTIFIER: US 6003067 A
** See image for Certificate of Correction **
TITLE: Data transmission controlling method and data transmission controlling
system, and computer memory product

Application Filing Date (1):
19970623

Detailed Description Text (13):
FIG. 8 is a diagram showing such a condition transition for the processing
procedures. As described above, the screen data transmission has been demanded from
the server 10A at the time point T1, and then, the hidden area is detected, and the
hidden area is notified to a client 30A at the time point T2. In the client 30A,
the screen data except screen data corresponding to the hidden area is extracted.
The screen data extracted at the time point T3 is transmitted to the server 10A and
is displayed.

Detailed Description Text (26):
As shown in FIG. 11, at a time point T13, the client 30B stores in the table 33b
the hidden area notified from the server 10B, so as to transmit the extracted
screen data to the server 10B (at step S31). As shown in FIG. 12, the server 10B
receives the screen data from the client 30B, so as to display it on the screen (at
step S41).

Detailed Description Text (27):
The client 30B measures (at step S32) with a timer 41 the setting time stored in
the set time management table 41, after the lapse of the predetermined time period
(at step S33), so as to detect (at step S34) whether or not the screen image of the
client 30B has been changed by the screen change detecting unit 39. When the screen
image has been changed, the flag of the table 33b is erected (at step S36). When
the screen image has not been changed, without erection of the flag at a time point
T14, the flag data is notified to the server 10B (at step S36).

Detailed Description Text (28):
At a time point T14, the server 10B which has received the flag data moves (at step
S42), the window of the client 30B, having notified the flag data, to the uppermost
layer to cause a voice or a message from a speaker 23 to speak, so as to notify (at
step S43) of the client number. Thus, the user of the server 10B can recognize
which client does not change the screen data within the predetermined period. The
server 10B demands (at step S44) the screen data of the client 30B at a time point
T15. The client 30B receives the demand from the server 10B and extracts the screen
data as in the first embodiment at a step S31, so as to transmit it to the server
10B. The server 10B displays the transmitted screen data (at step S45).

Detailed Description Text (29):
When the screen of the client in the predetermined time is not changed, since that
is notified to the server by such a data transmission controlling operation, the
user of the server can know promptly that which client has not changed the screen
data. Although the server 10B tells of the client after the window has been caused
to move to the uppermost layer at a step S42 and at a step S43, the operation is

not limited to them only. Only either of them will do. Also, the notification of
the client can be conducted through not only speaking of the voice or the message,
but also, processing such as window blinking, window discoloration, etc.

Detailed Description Text (30):
FIG. 11 and FIG. 12 describe the processing procedures in the notifying operation
of the screen image change from the client to the server, which is not the only
procedure. For example, the notification of the screen image change may be demanded
from the server to the client. The processing procedures in this case will be
described hereinafter.

Detailed Description Text (31):
FIG. 13 is a flow chart showing the processing procedures for a server when the
notification of the screen image change is demanded from the server to the client.
FIG. 14 is a flow chart showing the processing procedures of a client in this case.
When the windows are superposed and displayed on the server screen, the data
transmission processing operation is conducted as in the first embodiment with the
description thereof being omitted.

Detailed Description Text (32):
As shown in FIG. 14, at a time point T23, the client 30B stores in the table 33b
the hidden area notified from the server 10B to transmit (at step S61) the
extracted screen data to the server 10B. As shown in FIG. 13, the server 10B
receives the screen data from the client 30B and displays (at step S51) it on the
screen. The predetermined time set in the predetermined client is measured with the
timer 22 based on the set time management table 20 (at step S52). The timer 22 has
the table 22a and can measure the respective setting times of a plurality of
clients. After the lapse of the predetermined time (at step S53), at a time point
T24, the flag data of the area data management unit 33 which the client has, is
demanded (at step S54).

Detailed Description Text (36):
When no changes have been conducted within the predetermined time on the screen of
the client by the aforementioned data transmission controlling operation, the user
of the server can know as soon as possible which client does not change the screen
data. Since the window of the client where the screen image has been changed is
caused to move to the uppermost layer, the server can react quickly to the screen
image change of the client. Further, since the respective setting times by a
plurality of clients are measured on the server, and the notification of the change
in the respective screen data is demanded by the server, the controlling operation
of the window displaying operation is likely to be conducted about a plurality of
clients.

Detailed Description Text (40):
The client 30C has an area data management unit 33, a voice recognition unit 42, a
button detecting unit 44 and a recognition data registration unit 43. The other
configurations of the client 30C are similar to those of the first embodiment. The
description thereof will be omitted, because the same reference numerals are given
to the same units. The area data management unit 33 is provided with a table 33b
having a flag. A flag indicating the existence of the screen data change and a
hidden area are stored in the table 33b and the flag is erected when the screen
data has been changed. When the voice recognition unit 42 recognizes the voice
which the user of the client 30C has spoken and the voice is a voice registered in
the recognition data registration unit 43, it judges that the user of the client
30C has shown some action, so as to notify the server 10C of it. The button
detecting unit 44 detects that a question button provided on the window of the
client has been operated and judges that the user of the client 30C has shown some
action, so as to notify the server 10C of it.

Detailed Description Text (42):

As shown in FIG. 16, at a time point T33, the client 30C stores in the table 33b the hidden area notified from the server 10C, so as to transmit the extracted screen data to the server 10C (at step S71). When the voice from the user is detected (at step S72) by the voice recognition unit 42 after the screen data has been transmitted, it is judged (at step S73) whether or not the detected word is a word, registered in the recognition data registration unit 43, such as "question", "Excuse me" or the like. When the word is a registered word, it is notified to the server 10C at a time point T34 (at step S74). Then, the screen image changes are detected (at step S75) by the screen change detection unit 39. When the screen has been changed, the flag of the area data management unit 33 is erected as in that of the second embodiment. When the screen has not been changed, the flag is not erected.

Detailed Description Text (43):
At a step S72 and a step S73, a processing operation may be conducted to judge whether or not an operation of a question button has been detected by the button detecting unit 44, instead of voice to be recognized. By the operation of the question button, it is notified to a server 10C that the user of the client has shown an action. The operation of the question button cannot always be detected. Furthermore, whether or not the user of the client has operated a completion button, when the input processing operation has been completed, can be detected. The operation can be processed to notify the server 10C of it.

Detailed Description Text (44):
The server 10C having received the notification of the voice from the client 30C makes the voice or the message speak as shown in FIG. 17, to notify of a client number (at step S81). Thus, the user of the server 10C can recognize the user of which client 30C has shown the action. The server 10C demands (at step S82) the screen data to the client 30C shown some action at a time point T35. When the screen image changes after the screen data is demanded by the server 10C (at step S76), the client 30C transmits the screen data (at step S77). When the screen image is not changed (at step S76), the screen data is not transmitted. The server 10C judges (at step S83) whether or not the screen data has been transmitted from the client 30C. When the screen data has been transmitted, the display of the screen data is renewed (at step S84). When the screen data has been transmitted, the displaying of the screen data can be renewed and also, the window of the client can be moved to the uppermost layer.

Current US Cross Reference Classification (3):
715/790

HyperText Transfer Protocol (HTTP)--Within a Web-based environment, Web servers transfer HTML pages to clients using HTTP. HTTP can be thought of as a lightweight file transfer protocol optimized for transferring small files. HTTP reduces the inefficiencies of the FTP protocol. HTTP runs on top of TCP/IP and was developed specifically for the transmission of hypertext between client and server. The HTTP standard is changing rapidly. Secure Hypertext Transfer Protocol (S-HTTP)--a secure form of HTTP, mostly for financial transactions on the Web. S-HTTP has gained a small level of acceptance among merchants selling products on the Internet as a way to conduct financial transactions (using credit card numbers, passing sensitive information) without the risk of unauthorized people intercepting this information. S-HTTP incorporates various cryptographic message formats such as DSA and RSA standards into both the Web client and the Web server. File Transfer and Access Management (FTAM)--The OSI (Open Systems Interconnection) standard for file transfer, file access, and file management across platforms.

Detailed Description Text (607):
The following product provides File Transfer translation: IBM's Files On-Demand gateway--acts as a gateway between Web-based and mainframe-based FTP services to allow users to download mainframe-based files from a World Wide Web browser.

Detailed Description Text (654):
X.400--The X.400 message handling system standard defines a platform independent standard for store-and-forward message transfers among mail servers. X.400 is often used as a backbone e-mail service, with gateways providing interconnection with end-user systems.

Detailed Description Text (655):
SMTP--Simple Mail Transfer Protocol (SMTP) is a UNIX/Internet standard for transferring e-mail among servers.

Detailed Description Text (758):
Encryption that is architected into Web-based solutions Netscape's Secure Sockets Layer (SSL)--provides encryption for World Wide Web browsers. S-HTTP--a secure version of the HTTP data transfer standard; used in conjunction with the World Wide Web.

Detailed Description Text (2260):
When the Server has retrieved all of the data meeting the search criteria, the Server builds the last "page." When the last page is returned to the client, the "last found key" is left blank. This notifies the client the search is complete and no more data matching the search exists on the Server. Note that the last page is usually smaller than the other pages.

Detailed Description Text (2642):
The examples have shown a single exception handler being used. In practice it is more likely that multiple will be used. For example, the exception handler on a server may have different requirements or constraints than a client, or one client may be GUI based and display pop-up error messages, where another client is a batch program that needs to send notification messages to Operations. This can be handled by creating multiple handlers or using the Strategy pattern to customize the behavior.

Previous Doc   Next Doc   Go to Doc#

☐ Generate Collection   Print

L27: Entry 1 of 3   File: USPT   Oct 21, 2003

DOCUMENT-IDENTIFIER: US 6636242 B2
TITLE: View configurer in a presentation services patterns environment

Detailed Description Text (495):
Possible Product Options TelAlert; E-mail Systems e-mail systems--some e-mail
systems and fax servers can be configured to generate pages to notify users when a
defined event occurs such as e-mail/fax arriving. Telamon's TelAlert--TelAlert
provides notification capabilities for UNIX systems. For example, it can page
support personnel in the event of system problems.

Detailed Description Text (2146):
Definitions Starting Key The Starting Key is the initial starting point for the
search. The database will begin searching for data (customers in the message trace
above) at the Starting Key. An example starting key could be "A*". Last Found Key
The Last Found Key is used to request subsequent pages of data from the Server and
the database. The "last found key" defines the starting point for the next data
request. The Server will begin searching for data at the "last found key" and
continue until it has retrieved a full "page" of information. When all of the data
has been retrieved from the Server and Database, the Last Found Key is left blank.
This notifies the Client that all the data has been sent. Intermediate Page An
intermediate "page" is returned for every request but the last. When a client
receives an intermediate page and a "last found key", the client knows more "pages"
of data exist on the server. In order to obtain an intermediate "page," a "last
found key" must be passed from the client to the server. When the Server has
retrieved a full "page" of data, the new "last found key" is saved. It is then
passed back with the intermediate "page." The new "last found key" defines the
starting point for the next data request. Last Page When the Server has retrieved
all of the data meeting the search criteria, the Server builds the last "page."
When the last page is returned to the client, the "last found key" is left blank.
This notifies the client the search is complete and no more data matching the
search exists on the Server. Note that the last page is usually smaller than the
other pages. Empty Page When no data are selected from the search criteria, the
server builds an empty page signaling to the client no more data exist on the
server. Static or Dynamic Page Size The page size can be defined statically or
dynamically. The message trace diagram in FIG. 99 depicts a static page size. If
you'd like a dynamic page size, the client must pass an additional parameter with
each request to the Server. The additional parameter would be the page size. The
steps associated with FIG. 99 will now be set forth.

Detailed Description Text (2489):
The examples have shown a single exception handler being used. In practice it is
more likely that multiple will be used. For example, the exception handler on a
server may have different requirements or constraints than a client, or one client
may be GUI based and display pop-up error messages, where another client is a batch
program that needs to send notification messages to Operations. This can be handled
by creating multiple handlers or using the Strategy pattern to customize the
behavior.

L27: Entry 2 of 3   File: USPT   Jul 8, 2003

DOCUMENT-IDENTIFIER: US 6590588 B2
TITLE: Wireless, radio-frequency communications using a handheld computer

Detailed Description Text (378):
Another possible attack is for someone to impersonate the base station 170 and
proxy server 180. The attacking rogue server would attempt to force the wireless
client 405 to accept a new public key as part of the public key rejection mechanism
outlined above in step number 7 above. In order for this attack to be successful,
however, the rogue server must know the private key of the real proxy server 180.
Furthermore, the rogue server must be able to receive and transmit messages using
the unique identification number of the real proxy server 180. Thus, although an
attack premised on impersonation of a base station 170 and a proxy server is
possible, such an attack would be very difficult to mount. To further reduce the
risk of this attack, the wireless client 405 software asks user permission through
a dialog before accepting a new public key from the proxy server 180. Users are
forewarned, through means other than the wireless network (e.g., wireline e-mail,
or hard copy delivery) when a proxy server 180 public key is changed so that
"legal" changes to the proxy server 180 public key do not come as a surprise to a
user. Because the user knows of any legal change to the proxy server 180 public key
before the change is made, base station 170 and proxy server 180 impersonation
attacks can be defeated by user denial of permission to use new public keys that
are not accompanied by appropriate user notification.

Detailed Description Text (461):
When RMP is being used over a network that does not guarantee delivery of packets,
RMP provides a mechanism for the re-transmission of lost packets. Most reliable
protocol designs rely on acknowledgements from the remote host to indicate to the
sender that a packet was properly received. Then, if an acknowledgement is not
received within a specified timeout period, the packet is resent. This method is
not used in RMP because it forces a minimum of three packets to be exchanged for a
single transaction (request to proxy server 180, response to wireless client 405,
acknowledgement of response to proxy server 180).